

dunetpc - Bug #23510

Dataprep segfaults on long time windows

11/04/2019 10:51 AM - David Adams

Status:	Closed	Start date:	11/04/2019
Priority:	Normal	Due date:	
Assignee:	David Adams	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
I see segfault in dataprep when I try to process APA7 run 10157 which was taken with 100 ms (200k tick) time window.			

History

#1 - 11/04/2019 11:03 AM - David Adams

In debugger, I see crash is in `raw::Uncompress` which is being called with a too-small output container. I don't know why we have to pre-size that vector but the decompression expects this.

That size is taken from `digit.Samples()` which returns unsigned short (max 65k) which is not sufficient for these long time windows. This is in `dunetpc/dune/DataPrep/Tool/AcdDigitReader_tool.cc`.

I am modifying that tool to directly copy the compressed ADC vector (`digit.ADCs()`) in the case that the compression is off (`digit.Compression() == raw::kNone`) and to otherwise log a warning message if the uncompressed size is smaller than the compressed size.

I assume we are not compressing `RawDigit` for `protoDune` or `coldbox` data. We should be careful not to compress MC data when the time window is too long. And, IMHO, `RawDigit` should be checking that no one asks for that.

#2 - 11/04/2019 01:40 PM - Thomas Junk

I am putting in a much-belated request to the LArSoft coordination meeting to enlarge `fSamples` to a `ULong64_t`. It broke a line or two of code in `dunetpc` (old 35-ton method), and perhaps other experiment code, so it needs to be rolled out carefully, but this is long overdue.

There is a `fcl` parameter for `PDSPTPCRawDecoder_module` that asks for the data to be re-compressed using the "Huffman" method in `lardataobj/RawData/raw.cxx`. `PDSPDataInterface_tool.cc` program lacks this option as it was never used (except once by me last week when trying to decode one of these very long events).

The solution proposed above sounds safe, though one can imagine an overflowed `Samples()` still being larger than the compressed NADC and still being wrong. So far we only compress the MC raw digits, and MC doesn't have more than 16k ticks (that I know of).

#3 - 11/04/2019 02:01 PM - David Adams

I made the above change and verified I can now read long window data events directly from raw fragment data (where `RawDigit` is not compressed). Tom points out that we may want or have to compress the `RawDigits` if we write them in an intermediate step.

Change is committed to `dunetpc`.

#4 - 12/26/2019 12:19 PM - David Adams

- Status changed from Assigned to Closed

I suppose this is resolved.